NASA-CR-201068

111
65646

RIACS

Research Institute for Advanced Computer Science
NASA Ames Research Center

An all-at-once reduced Hessian SQP scheme for aerodynamic design optimization

Dan Feng and Thomas H. Pulliam

RIACS Technical Report 95.19 October 1995

An all-at-once reduced Hessian SQP scheme for aerodynamic design optimization

Dan Feng and Thomas H. Pulliam

The Research Institute for Advanced Computer Science is operated by Universities Space Research Association, The American City Building, Suite 212, Columbia, MD 21044 (410)730-2656

Work reported herein was supported in part by NASA under contract NAS 2-13721 between NASA and the Universities Space Research Association (USRA).

AN ALL-AT-ONCE REDUCED HESSIAN SQP SCHEME FOR AERODYNAMICS DESIGN OPTIMIZATION*

DRAFT

by

Dan Feng[†] *and Thomas H. Pulliam*[‡]

*Work reported herein was supported by NASA under contract NAS 2-13721 between NASA and the Universities Space Research Association (USRA).

[†]Research Institute for Advanced Computer Science (RIACS), Mail Stop T20G-5, NASA Ames Research Center, Moffett Field, CA 94035-1000, USA. (feng@riacs.edu)

[‡]Fluid Dynamics Division, Mail Stop T27B-1, NASA Ames Research Center, Moffett Field, CA 94035-1000, USA. (pulliam@nas.nasa.gov)

Abstract

This paper introduces a computational scheme for solving a class of aerodynamic design problems that can be posed as nonlinear equality constrained optimizations. The scheme treats the flow and design variables as independent variables, and solves the constrained optimization problem via reduced Hessian successive quadratic programming. It updates the design and flow variables simultaneously at each iteration and allows flow variables to be infeasible before convergence. The solution of an adjoint flow equation is never needed. In addition, a range space basis is chosen so that in a certain sense the "cross term" ignored in reduced Hessian SQP methods is minimized. Numerical results for a nozzle design using the quasi-one-dimensional Euler equations show that this scheme is computationally efficient and robust. The computational cost of a typical nozzle design is only a fraction more than that of the corresponding analysis flow calculation. Superlinear convergence is also observed, which agrees with the theoretical properties of this scheme. All optimal solutions are obtained by starting far away from the final solution.

Key words. design optimization, constrained optimization, reduced Hessian methods, quasi-Newton methods, successive quadratic programming

AMS(MOS) subject classification. 65K05, 90C06, 90C30, 90C31, 90C90

Abbreviated title. SCHEME FOR DESIGN OPTIMIZATION

1 Introduction

An aerodynamic design optimization problem can often be posed as

$$\begin{aligned} &\min \ \ I(X,u) \\ &\text{s.t.} \ \ F(X,u) = 0, \end{aligned} \tag{1.1}$$

where $X \in \Re^n$ denotes the discretized flow variables, and $u \in \Re^m$ denotes the design variables, which, for example, could be geometry parameters describing the shape of a profile; $I : \Re^{n+m} \to \Re$ is a cost function, which may, for example, measure the deviation from a desired surface pressure distribution; $F : \Re^{n+m} \to \Re^n$ is a discretized version of the governing equations of the flow field. It is often the case that I and F are nonlinear, and the number of flow variables is much larger than the number of design variables ($n >> m$).

Many computational methods for solving (1.1) have been developed, and each seeks to find the solution by solving a series of subproblems. One can categorize these methods by looking at how the design and flow variables are treated. In one category, design optimization approaches treat X as a dependent variable of u. Representatives in this category are the brute-force approach, the implicit gradient approach (e.g. [18]) and the adjoint equations approach (e.g. [14]). These approaches have a feasibility requirement that is satisfied by solving the nonlinear flow equations to convergence in each design cycle. They only differ in how $\nabla_u I$ is calculated at each iteration. The brute-force approach computes $\nabla_u I$ by one-side finite differences, which requires m solutions of the flow equations. The implicit gradient approach and the adjoint equations approach also require at least two solutions of nonlinear flow equations at each design iteration.

In another category, design optimization approaches treat X and u as independent variables. Given X_k, u_k, each approach in this category finds ΔX_k, Δu_k and updates all the variables at once such that $X_k + \Delta X_k$, $u_k + \Delta u_k$ better solves (1.1), and iterates until a satisfactory solution is obtained. One interesting property is that the flow equations are not required to be satisfied until convergence. Approaches in this category are expected to mitigate the computational cost of repeatedly solving the nonlinear flow equations required by other methods. The success of flow solvers based on Newton's method (e.g. see Barth [1] and [13]) also makes the all-at-once approach more tractable.

The approach introduced in this paper belongs to this latter category. Methods that simultaneously update design and flow variables have been studied by many authors including Frank and Shubin [7], Huffman et al [13], and Hou et al [12]. The scheme in [13] is based on Successive Quadratic Programming (SQP) for equality constrained optimization. In their scheme the Hessian matrix is approximated by selectively calculating some second order terms instead of using an update formula. The calculation of second order terms could be prohibitive in many situations. The formulation introduced in [12] is based on successive linear programming. This scheme may suffer from convergence problems and in addition a linear adjoint system has to be solved at each iteration. Frank and Shubin [7] compare three optimization-based methods, including an all-at-once scheme (based on the full Hessian SQP that would be too expensive to form for large problems) for solving aerodynamic design problems. They point out, however, that an all-at-once scheme could be very efficient if carefully implemented.

SQP is a mature and successful technique for solving nonlinear constrained optimization problems due to its relatively low computational cost and fast convergence. However, it is not widely used in aerodynamic design optimization. The major obstacle is that aerodynamic design optimization problems are often very large. In order to have an efficient and robust implementation of SQP methods, one has to take full advantage of special characteristics of the aerodynamic design problems. This paper is intended to address some aspects of this issue.

Since the approach introduced in this paper originates from SQP for equality constrained optimization, we first describe the full Hessian SQP scheme in the context of (1.1).

At each iteration of SQP methods, a quadratic approximation to the Lagrangian function of (1.1)

$$L(X, u, \lambda) = I(X, u) + \lambda^T F(X, u),$$

is minimized subject to a linearization of the flow equations. This gives a subproblem

$$\min_{d \in \mathbb{R}^{(n+m)}} g_k^T d + \tfrac{1}{2} d^T H_k d \tag{1.2}$$

$$\text{subject to } F_k + A_k^T d = 0, \tag{1.3}$$

where $d = \begin{pmatrix} \Delta X \\ \Delta u \end{pmatrix}$, $g_k = \begin{pmatrix} \frac{\partial I}{\partial X} \\ \frac{\partial I}{\partial u} \end{pmatrix}$ at (X_k, u_k), $A_k = \begin{pmatrix} \frac{\partial F}{\partial X} \\ \frac{\partial F}{\partial u} \end{pmatrix}$ at (X_k, u_k), $F_k = F(X_k, u_k)$, and H_k is the Hessian (or approximation) of the Lagrangian function.

One difficulty implementing full SQP methods for aerodynamic design optimization is that the analytic Hessian of the Lagrangian is usually hard to obtain. Another difficulty is that approximating the Hessian by update schemes tends to create large dense matrices ($(n+m) \times (n+m)$).

These difficulties can be overcome by dropping certain non-critical second order information. This idea has been pursued by many researchers including Coleman and Conn [4], Nocedal and Overton [16], Byrd and Nocedal [3]. We illustrate the basic idea as follows.

One way of solving (1.2-1.3) is via separation of variables. Suppose we can compute two matrices $Y_k \in \Re^{(n+m)\times n}$ and $Z_k \in \Re^{(n+m)\times m}$ such that the matrix $[Y_k \;\; Z_k]$ is nonsingular and $Z_k^T A_k = 0$ (Z_k is a basis for the null space of A_k^T). Let $d = Y_k d_1 + Z_k d_2$ and plug this into (1.2-1.3). If we assume $Z_k^T H_k Z_k$ is positive definite (which is reasonable because this matrix is indeed positive definite close to the solution due to the second order sufficient optimality condition), by standard techniques, d_1 and d_2 are given by

$$\begin{aligned} d_1 &= -(A_k^T Y_k)^{-1} F_k \\ d_2 &= -(Z_k^T H_k Z_k)^{-1}(Z_k^T g_k + Z_k^T H_k Y_k d_1). \end{aligned}$$

To avoid the computation of H_k, the "cross" term $Z_k^T H_k Y_k d_1$ is simply ignored and $Z_k^T H_k Z_k$ is approximated by an $m \times m$ matrix B_k using a variable metric formula such as BFGS. Consequently,

$$d_1 = -(A_k^T Y_k)^{-1} F_k \tag{1.4}$$

$$d_2 = -B_k^{-1} Z_k^T g_k. \tag{1.5}$$

Methods based on (1.4-1.5) are referred to as reduced Hessian SQP methods.

The biggest advantage of reduced Hessian SQP methods for aerodynamic design optimization is that only a small $m \times m$ matrix holding second order information needs to be maintained and updated. Theoretically, reduced Hessian SQP methods are not as effective as full Hessian SQP methods. However, many versions are shown to have 2-step superlinear local convergence (for example see [16], [3] and [19]), which is good enough for most applications.

One challenge of implementing (1.4-1.5) for aerodynamic design optimization is the choice of Z_k and Y_k. Traditionally, Z_k and Y_k are chosen from an orthonormal matrix obtained from the QR factorization of A_k. QR factorization based reduced Hessian SQP algorithms are proven to be very robust in practice (for example, see [11]). However, for aerodynamics design optimization, A_k contains the flow Jacobian matrix, which is usually very large. Orthonormal base are usually too expensive to compute.

Many authors, including Gabay [8], Gilbert [9], Fletcher [6], and Xie and Byrd [19], have proposed more economical choices of Z_k and Y_k which are not orthogonal. One popular choice of Z_k is

$$Z_k = \begin{pmatrix} -(\frac{\partial F}{\partial X})^{-T}(\frac{\partial F}{\partial u})^T \\ I \end{pmatrix} = \begin{pmatrix} -S \\ I \end{pmatrix}, \tag{1.6}$$

where, in case of aerodynamic design optimization, S is the sensitivity matrix. It can easily be verified that $Z_k^T A_k = 0$. The calculation of the sensitivity matrix is usually affordable for many applications. An economical choice of Y_k is $[I \ 0]^T$, which is used by many authors including Biegler *et al* [2], and Dennis *et al* [5].

This paper introduces a reduced Hessian SQP scheme for the aerodynamic design problem (1.1). Equations (1.4-1.5) are used to compute a trial step at each iteration. The popular choice of Z_k given by (1.6) is used. However, a different choice of Y_k is considered. We show that using $[I \ 0]^T$ has an undesirable effect of potentially producing a larger cross term. Here Y_k is chosen such that $Z_k^T Y_k = 0$ holds, which in a certain sense will minimize the cross term. We also show that this choice of Y_k has the advantage of gaining accuracy in the reduced Hessian approximation, which is very important for ensuring the reliability and robustness of a reduced Hessian SQP scheme.

The new scheme is implemented and tested on the shape design of a transonic nozzle. The transonic flow through the nozzle with a given area ratio is governed by the quasi-one-dimensional Euler equations. The transonic conditions which generate a shock within the nozzle present a difficult test case, where methods typical of practical aerodynamic applications are required. Such methods include, finite difference, finite element, and unstructured grid finite volume techniques employing various forms of highly nonlinear algorithm constructions.

This paper is organized as follows. First the all-at-once reduced Hessian SQP scheme is introduced in Section 2. The solution of the quasi-one-dimensional Euler equations for flow through a nozzle with a given area ratio is discussed in Section 3. Section 4 first describes the design optimization problem used in our testing, then provides computational results and the performance evaluation of the new scheme on the design problem. The testing is carried out by using different numbers of design variables (spline coefficients for the nozzle shape) for the test problem. Finally, we give some concluding remarks in Section 5.

2 An all-at-once reduced Hessian SQP scheme

Let B_k be an approximate to $Z_k^T H_k Z_k$. Let Z_k be given by (1.6). We define Y_k by

$$Y_k = \begin{pmatrix} I \\ S^T \end{pmatrix}. \tag{2.1}$$

Clearly $Z_k^T Y_k = 0$ holds.

Let $d = Y_k d_1 + Z_k d_2$. Replacing the reduced Hessian matrix by B_k and ignoring the cross term in (1.5), we have

$$d_2 = -B_k^{-1} Z_k^T g_k.$$

From (1.4), d_1 satisfies $(A_k^T Y_k) d_1 = -F_k$, which is equivalent to

$$\left[\left(\frac{\partial F}{\partial X} \right)^T + \left(\frac{\partial F}{\partial u} \right)^T S^T \right] d_1 = -F_k,$$

which in turn is equivalent to

$$J_k(I + SS^T)d_1 = -F_k, \tag{2.2}$$

where $J_k = \left(\frac{\partial F}{\partial X}\right)^T$ at (X_k, u_k) is the current Jacobian matrix of the flow equation. Equation (2.2) can be solved by first solving $J_k y = -F_k$ for y, and then solving $(I + SS^T)d_1 = y$ for d_1. The solution of the former is simply the Newton step calculation from the flow equations at the current iteration, which can be provided by any Newton's method based flow solver. The solution of the latter can be obtained by the conjugate gradient method, which is guaranteed to converge within $(m + 1)$ iterations due to $Rank(SS^T) = m$ (see Golub and Van Loan [10]). Another way of solving $(I + SS^T)d_1 = y$ is by inverting $(I + SS^T)$ directly. It is easy to show that

$$(I + SS^T)^{-1} = I - S(I + S^T S)^{-1} S^T.$$

Note that $(I + S^T S)$ is only an $m \times m$ matrix and its factorization can be obtained at minimal cost.

After d_1 and d_2 are available, d is given by

$$\begin{aligned} d &= Y_k d_1 + Z_k d_2 \\ &= \begin{pmatrix} I \\ S^T \end{pmatrix} d_1 + \begin{pmatrix} -S \\ I \end{pmatrix} d_2. \end{aligned}$$

Afterwards, we update our solution $\begin{pmatrix} X_{k+1} \\ u_{k+1} \end{pmatrix} = \begin{pmatrix} X_k \\ u_k \end{pmatrix} + \alpha d$, update B_k and the Langrange multiplier λ_k (which is used implicitly in the merit function calculation, see below), and go to next iteration.

The Lagrange multiplier is asked to satisfy

$$(Y_k^T A_k)\lambda_k = -Y_k^T g_k, \tag{2.3}$$

which is equivalent to

$$(I + SS^T)(J^T \lambda_k) = -Y_k^T g_k.$$

As a matter of fact, we will see that our scheme only needs the value of $(J^T \lambda_k)$.

We establish the following lemma that is useful to us.

Lemma 2.1 *Let M, $N \in \Re^{n \times m}$ with $n \geq m$. If M and N have the same range space of dimension m, then*

$$M(N^T M)^{-1} N^T = M(M^T M)^{-1} M^T. \tag{2.4}$$

Proof. Clear (N^TM) is invertible. Since M and N have the same range space,

$$M(M^TM)^{-1}M^T = N(N^TN)^{-1}N^T. \tag{2.5}$$

Postmultiplying each side by $M(N^TM)^{-1}N^T$ and using (2.5) gives

$$\begin{aligned} & M(M^TM)^{-1}M^TM(N^TM)^{-1}N^T = N(N^TN)^{-1}N^TM(N^TM)^{-1}N^T \\ \Longrightarrow \quad & M(N^TM)^{-1}N^T = N(N^TN)^{-1}N^T \end{aligned}$$

which gives (2.4) using (2.5). □

When certain update criteria are satisfied, the Hessian approximation update is carried out via the BFGS formula

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}, \tag{2.6}$$

where

$$y_k = Z_{k+1}^T[\nabla_{(X,u)}L(X_{k+1}, u_{k+1}, \lambda_k) - \nabla_{(X,u)}L(X_k, u_k, \lambda_k)] \tag{2.7}$$

$$s_k = (Z_{k+1}^T Z_{k+1})^{-1} Z_{k+1}^T \alpha d, \tag{2.8}$$

with α being the step length resulted from a line search. This choice of y_k, s_k pair is among the choices recommended by Nocedal and Overton [16]. From the definition of the Lagrangian function

$$\begin{aligned} y_k &= Z_{k+1}^T[(g_{k+1} + A_{k+1}\lambda_k) - (g_k + A_k\lambda_k)] \\ &= Z_{k+1}^T[g_{k+1} - (g_k - A_k(Y_k^TA_k)^{-1}Y_k^Tg_k)] && (2.9) \\ &= Z_{k+1}^T[g_{k+1} - (I - Y_k(Y_k^TY_k)^{-1}Y_k^T)g_k] && (2.10) \\ &= Z_{k+1}^T[g_{k+1} - Z_k(Z_k^TZ_k)^{-1}Z_k^Tg_k]. && (2.11) \end{aligned}$$

Equation (2.9) to (2.10) is due to Lemma 2.1. Equation (2.10) to (2.11) is because

$$Z_k(Z_k^TZ_k)^{-1}Z_k^T + Y_k(Y_k^TY_k)^{-1}Y_k^T = I$$

due to $Z_k^TY_k = 0$.

To ensure convergence, a merit function is needed to monitor the progress towards the solution. We choose to use the l_1 merit function for its simplicity and low computational cost. The l_1 merit function is defined as

$$\phi_\mu(X, u) = I(X, u) + \mu\|F(X, u)\|_1.$$

The directional derivative of ϕ_μ along d is given by

$$D_{\phi_{\mu_k}}(X, u; d) = g_k^Td - \mu_k\|F_k\|_1. \tag{2.12}$$

Plugging $d = Y_k d_1 + Z_k d_2$ into (2.12) yields

$$D_{\phi_{\mu_k}}(X, u; d) = g_k^T Y_k d_1 + g_k^T Z_k d_2 - \mu_k \|F_k\|_1.$$

Clearly, on the one hand

$$g_k^T Z_k d_2 = -g_k^T Z_k B_k^{-1} Z_k^T g_k \leq 0, \tag{2.13}$$

since B_k is forced to be positive definite. On the other hand, from (2.3) and $d_1 = -(Y_k^T A_k)^{-1} F_k$,

$$g_k^T Y_k d_1 = \lambda_k^T F_k = (J^T \lambda_k)^T (J^{-1} F_k). \tag{2.14}$$

Note that both $(J^T \lambda_k)$ and $(J^{-1} F_k)$ are available from previous calculations. From (2.13-2.14)

$$D_{\phi_{\mu_k}}(X, u; d) < 0$$

can be guaranteed by choosing $\mu_k > (\lambda_k^T F_k)/\|F_k\|_1$

We point out that J^T is not needed in the entire scheme, which is desirable in aerodynamic calculations.

One feature of our choice of Y_k is that the cross term ignored in the reduced Hessian SQP formulation is likely to be minimized. Since this is not a obvious claim, We establish the following lemma.

Lemma 2.2 *For a given matrix $N \in \Re^{n \times m}$ and a vector $p \in \Re^m$, over all the choices of matrices $M \in \Re^{n \times m}$ such that $(N^T M)$ is invertible, $\|M(N^T M)^{-1} p\|$ is minimized if M has the same range space as N.*

Proof. Let $N = UDV^T$ be the singular value decomposition of N with $U \in \Re^{n \times m}$ and $D, V \in \Re^{m \times m}$, and $M = QR$ be the QR decomposition of M with $Q \in \Re^{n \times m}$ and $R \in \Re^{m \times m}$. Clearly, D and R have full rank. Then

$$\begin{aligned} \|M(N^T M)^{-1} p\| &= \|QR(VDU^T QR)^{-1} p\| = \|Q(U^T Q)^{-1} D^{-1} V^T p\| \\ &= \|(U^T Q)^{-1} D^{-1} V^T p\| \geq \|D^{-1} V^T p\| / \sigma_{max}, \end{aligned} \tag{2.15}$$

where σ_{max} is the largest singular value of $(U^T Q)$. Since U and Q are unitary, $\|U^T Q\| \leq \|U^T\| \|Q\| = 1$, which implies $\sigma_{max} \leq 1$. Hence

$$\|M(N^T M)^{-1} p\| \geq \|D^{-1} V^T p\|.$$

The equality is achieved when $U^T Q$ is unitary (in this case $\sigma_{max} = 1$. Hence it is sufficient to show that $U^T Q$ is unitary if M and N have the same range space.

Clearly if M and N have the same range space, then Q and U have the same range space. Since Q and U are unitary, $QQ^T = UU^T$ holds. Therefore

$$(U^T Q)^T (U^T Q) = Q^T (UU^T) Q = Q^T (QQ^T) Q = (Q^T Q)(Q^T Q) = I,$$

which implies that U^TQ is unitary. □

Recall that $d_1 = -(A_k^TY_k)^{-1}F_k$. Choosing Y_k such that $Z_k^TY_k = 0$ implies that Y_k and A_k have the same range space due to $Z_k^TA_k = 0$. Hence by Lemma 2.2, $\|Y_kd_1\|$ achieves minimum value when $Z_k^TY_k = 0$. For a given Z_k and H_k, a smaller $\|Y_kd_1\|$ is desirable since it has a major contribution to the magnitude of the cross term $Z_k^TH_kY_kd_1$ ignored in the reduce Hessian formulation.

Choosing Y_k such that $Z_k^TY_k = 0$ also has the potential advantage of improving accuracy in the Hessian approximation. Note that (2.8) implies

$$Z_{k+1}s_k = Z_{k+1}(Z_{k+1}^TZ_{k+1})^{-1}Z_{k+1}^T\alpha d.$$

The Taylor expansion of y_k used in the step secant update gives

$$\begin{aligned}
y_k &= Z_{k+1}^T\nabla^2_{(X,u)(X,u)}L(X_k,u_k,\lambda_k)\alpha d + O(\|\alpha d\|^2) \\
&= Z_{k+1}^T\nabla^2_{(X,u)(X,u)}L(X_k,u_k,\lambda_k)Z_{k+1}(Z_{k+1}^TZ_{k+1})^{-1}Z_{k+1}^T\alpha d \\
&\quad +Z_{k+1}^T\nabla^2_{(X,u)(X,u)}L(X_k,u_k,\lambda_k)(I - Z_{k+1}(Z_{k+1}^TZ_{k+1})^{-1}Z_{k+1}^T)\alpha d \\
&\quad +O(\|\alpha d\|^2) \\
&= Z_{k+1}^T\nabla^2_{(X,u)(X,u)}L(X_k,u_k,\lambda_k)Z_{k+1}s_k \\
&\quad +Z_{k+1}^T\nabla^2_{(X,u)(X,u)}L(X_k,u_k,\lambda_k)Y_{k+1}(Y_{k+1}^TY_{k+1})^{-1}Y_{k+1}^T\alpha d + O(\|\alpha d\|^2).
\end{aligned}$$

Since B_{k+1} is expected to approximate the reduced Hessian matrix, it is desirable to keep the size of the second term as small as possible. This goal can be partially achieved by controlling the size of $Y_{k+1}(Y_{k+1}^TY_{k+1})^{-1}Y_{k+1}^Td$. Using $d = Y_kd_1 + Z_kd_2$, we have

$$\begin{aligned}
&\|(Y_{k+1}(Y_{k+1}^TY_{k+1})^{-1}Y_{k+1}^T)d\| \\
&= \|(Y_k(Y_k^TY_k)^{-1}Y_k^T)d\| + O(\|d\|^2) \\
&= \|Y_kd_1 + (Y_k(Y_k^TY_k)^{-1}Y_k^T)Z_kd_2\| + O(\|d\|^2). \qquad (2.16)
\end{aligned}$$

Choosing Y_k such that $Z_k^TY_k = 0$ makes $\|Y_kd_1\|$ achieve its minimum value as analyzed above, and makes $\|(Y_k(Y_k^TY_k)^{-1}Y_k^T)Z_kd_2\|$ zero. Hence it is likely that this choice would yield the lowest value for (2.16).

3 Quasi-one-dimensional Euler equations

For our purposes here, we have chosen one popular form of central finite differences with nonlinear artificial dissipation applied to the quasi-1D Euler equations.

The quasi-1D Euler equations are

$$\mathbf{F}(\mathbf{Q}) = \partial_x\mathbf{E}(\mathbf{Q}) + \mathbf{H}(\mathbf{Q}) + \mathbf{D}(\mathbf{Q}) = 0 \quad 0.0 \le x \le 1.0 \qquad (3.1)$$

where

$$\mathbf{Q} = \begin{bmatrix} \rho \\ \rho u \\ e \end{bmatrix}, \quad \mathbf{E} = a(x)\begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(e+p) \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} 0 \\ -p\partial_x a(x) \\ 0 \end{bmatrix} \qquad (3.2)$$

with ρ (density), u (velocity), e (energy), $p = (\gamma - 1)(e - 0.5\rho u^2)$ (pressure), $\gamma = 1.4$ (ratio of specific heats), and $a(x) = (1. - 4.(1 - a_t)x(1 - x))$ (the nozzle area ratio), with $a_t = 0.8$. For a given area ratio and shock location (here $x = 0.7$) an exact solution can be obtained from the method of characteristics.

We choose one popular form of central finite differences to discretize these equations.

$$\begin{aligned} \partial_x q \approx \delta_x q_j = \frac{q_{j+1} - q_{j-1}}{2\Delta x} \quad & j = 1, \ldots, Jmax \\ \Delta x = 1.0/(Jmax - 1), \quad & u_j = u(j\Delta x) \end{aligned} \tag{3.3}$$

It is common practice and well known that artificial dissipation must be added to the discrete central difference approximations in the absence of any other dissipative mechanism, especially for transonic flows, see Pulliam[17].

For simplicity here, we use a constant coefficient dissipation of the form

$$D^4(\mathbf{Q}) = \nabla_x \left(\epsilon_j^{(4)} \Delta_x \nabla_x \Delta_x \mathbf{Q}_j \right) \tag{3.4}$$

with

$$\nabla_\xi q_j \;\; = \;\; q_j - q_{j-1}, \quad \Delta_\xi q_j = q_{j+1} - q_j \tag{3.5}$$

with a typical value of $\epsilon^{(4)} = \frac{1}{100}$.

Boundary condition at $j = 1$ and $j = Jmax$ are defined in terms of physical conditions (taken from exact solution values) and will be treated as Dirchlet (fixed conditions) for now.

The total system we shall solve is

$$\mathcal{F}(\mathbf{Q}) = \begin{cases} \delta_x \mathbf{E}(\mathbf{Q})_j - \mathbf{H}(\mathbf{Q})_j + D_j^4(\mathbf{Q}), & j = 1, \cdots, J_N \\ B(\mathbf{Q})_i = 0, & i = 0, J_N \end{cases} \tag{3.6}$$

The Jacobian matrix for the linear dissipation case of Eq.(5) can be written exactly as

$$\mathcal{A} = \frac{\partial \mathcal{F}}{\partial \mathbf{Q}} = \begin{pmatrix} [C] & [D] & [E] & & & & & & & \\ [B] & [C] & [D] & [E] & & & & & & \\ [A] & [B] & [C] & [D] & [E] & & & & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & & & & \\ & & [A]_j & [B]_j & [C]_j & [D]_j & [E]_j & & & \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & & & [A] & [B] & [C] & [D] & [E] & \\ & & & & & [A] & [B] & [C] & [D] & [E] \\ & & & & & & [A] & [B] & [C] & [D] \\ & & & & & & & [A] & [B] & [C] \end{pmatrix} \tag{3.7}$$

where the elements ($[I]$ is the 3×3 identity matrix) are defined as

$$\begin{aligned}
[A]_j &= \epsilon^{(4)}[I] \\
[B]_j &= -4\epsilon^{(4)}[I] - \frac{1}{2\Delta x}[AJ]_j \\
[C]_j &= 6\epsilon^{(4)}[I] + [CJ]_j \\
[D]_j &= -4\epsilon^{(4)}[I] + \frac{1}{2\Delta x}[AJ]_j \\
[E]_j &= \epsilon^{(4)}[I]
\end{aligned}$$

with

$$[AJ]_j = a(x)_j \begin{bmatrix} 0 & 1 & 0 \\ (\gamma-3)u^2/2 & -(\gamma-3)u & (\gamma-1) \\ [-\gamma\, u\, e/\rho + (\gamma-1)u^3] & \left[\gamma e/\rho - \frac{3}{2}(\gamma-1)u^2\right] & \gamma u \end{bmatrix}_j \tag{3.8}$$

and

$$[CJ]_j = -\partial_x a(x)_j \begin{bmatrix} 0 & 0 & 0 \\ (\gamma-1)u^2/2 & -(\gamma-1)u & (\gamma-1) \\ 0 & 0 & \end{bmatrix}_j \tag{3.9}$$

The $\partial_x a(x)_j$ term is done with central differences and there are some slight modifications to the dissipation terms near the boundaries.

4 Test results for a nozzle design problem

The nozzle design problem. We assume that a target velocity distribution u_j^*, is given for each computational grid. The design problem we are trying to solve is

> *Find y_i, $i = 1, \cdots, m$ (spline coefficients describing $a(x)$), such that $\frac{1}{2}\sum_{j=1}^{Jmax}(u_j - u_j^*)^2$ is minimized subject to (3.6) being satisfied.*

For our test examples, the breakpoints of the spline are evenly distributed in the interval $[0, 1]$.

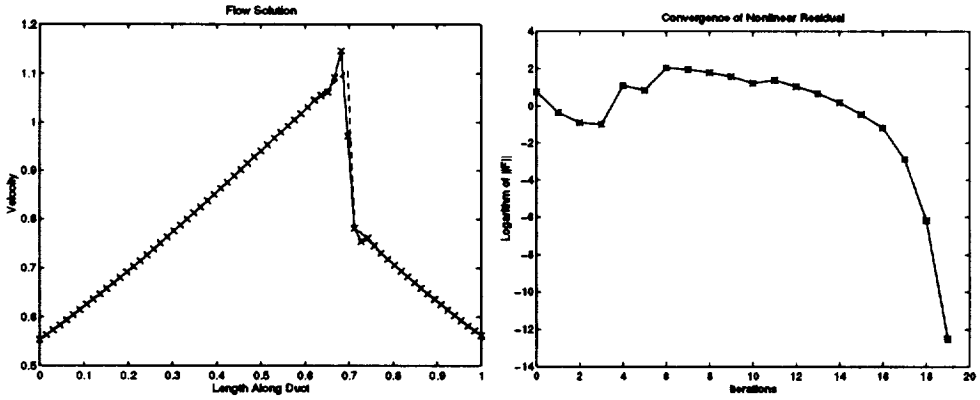
Some implementation issues. Each design is started from a flat nozzle and a flat flow solution. The first reduced Hessian approximation is given by $B_0 = Z_0^T Z_0$.

A trial step d is calculated as described in Section 2. A quadratic backtracking line search is carried out using the l_1 merit function, which yields a step length α_k. The following update criterion is used for the reduced Hessian approximation. At iteration k, B_k is updated using the BFGS formula (2.6) if

$$s_k^T y_k > 0.1\alpha_k \|Y_k d_1\|. \tag{4.1}$$

Otherwise, we set $B_{k+1} = B_k$. This criterion is devised to guard against loss in accuracy of B_k, as recommended by Xie and Byrd [19], and is critical for ensuring convergence of a reduced Hessian SQP algorithm.

We give test results on design problems with zero, two and ten design variables. Throughout this section the number of grids $Jmax$ is set consistently to 67. We chose this number since having an even breakpoint distribution for the splines requires that $(Jmax - 1)$ be divisible by the number of design variables plus one. Hence each problem has $67 \times 3 = 201$ flow variables, and the same number of constraint equations. The convergence was monitored by $\|N_k g_k\| + \|F_k\|$ ($\|F_k\|$ in case of zero design variables), where $N_k = Z_k(Z_k^T Z_k)^{-1} Z_k^T$ is a projector onto the null space. This quantity indicates how well the first order necessary optimality conditions are satisfied, and should be zero at the true solution. The tolerance is set to 10^{-7}.



(a) (b)

Figure 1: Computational results without design variables. (a) Target (dash line) and final (checked solid) flow solutions. (b) Convergence of norm of residual versus iterations.

Test results without design variables. These results are obtained by calling the flow solver with the optimal nozzle area ratio. We include these results as a baseline for measuring the cost of our optimization scheme. These results are also useful for examining the effect of the presence of design variables on the flow solution. The overshooting in the final solution can be controlled by nonlinear dissipation, as described in Pulliam [17].

Figure 1 (a) gives the final and target flow solutions. Figure 1 (b) shows the convergence history of the nonlinear flow residual versus the iterations. Figure 2 gives snapshots of partial solutions at various iterations. It can be seen that the transition is quite violent due to the existence of a shock in the solution.

Test results with two design variables. For this problem four spline coefficients are used. We keep the coefficients at the end points fixed, leaving the area ratios at the two between points as design variables. Figure 3 gives the test results for this case. Figure 3 (a) shows the initial, final and target flow solutions. Figure 3 (b) shows the initial, final and

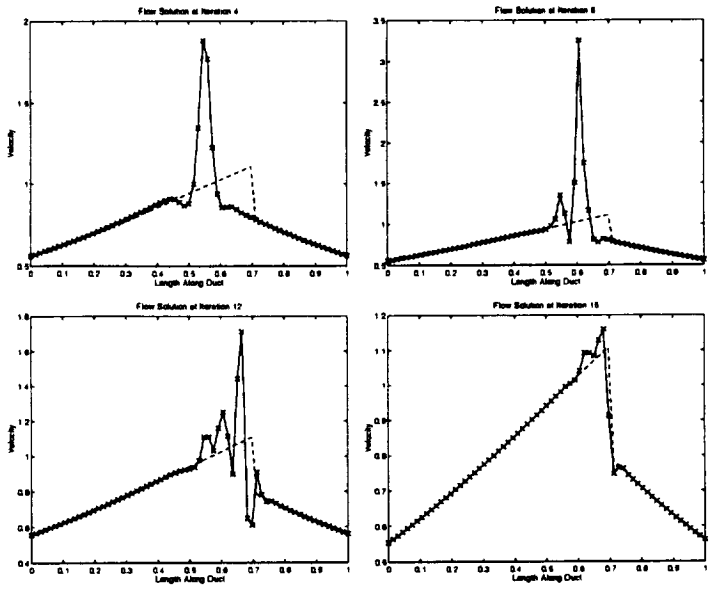


Figure 2: Snapshots of flow solutions at various iterations without design variables Dashed line – target solution; Checked solid line – current solution.

target nozzle solutions. Figure 3 (c) gives the convergence history of the objective function. Figure 3 (d) plots the convergence history of the quantities $\|N_k g_k\|$ and $\|F_k\|$.

The optimization converged quite rapidly taking 14 iterations. Figure 3 (a) and (b) show that the final flow and nozzle solutions match well to the corresponding target solutions. Figure 3 (c) shows that interates converged to a Kuhn-Tucker point (point that satisfies first order necessary conditions) superlinearly.

Test results with ten design variables. For this problem twelve spline coefficients are used. Again we keep the coefficients at the end points kept fixed, leaving the area ratios at the ten between points as design variables. Figure 4 gives the test results for this case.

It took 30 iterations for the optimization process to converge to the given tolerance. Figure 4 (a) and (b) show that the final flow and nozzle solutions match well to the corresponding target solutions.

The convergence is slower than the case with two design variables. We believe that this is partially due to over-parameterization that makes the design problem much harder to solve. Nevertheless, as shown in Figure 4 (d), it converged rapidly (superlinearly) to the solution after wandering around for a relatively longer transient period.

For the situation with two design variables, it was observed that flow solutions went through a less violent transient regime compared to the situation when running the flow solver without the presence of design parameters. For the situation with ten design variables, similar phenomena occurred. Figures 5 and 6 give some snapshots of intermediate flow and nozzle solutions with ten design variables at various intermediate iterations. These snapshots indicate that the transient is much milder with the presence of design variables.

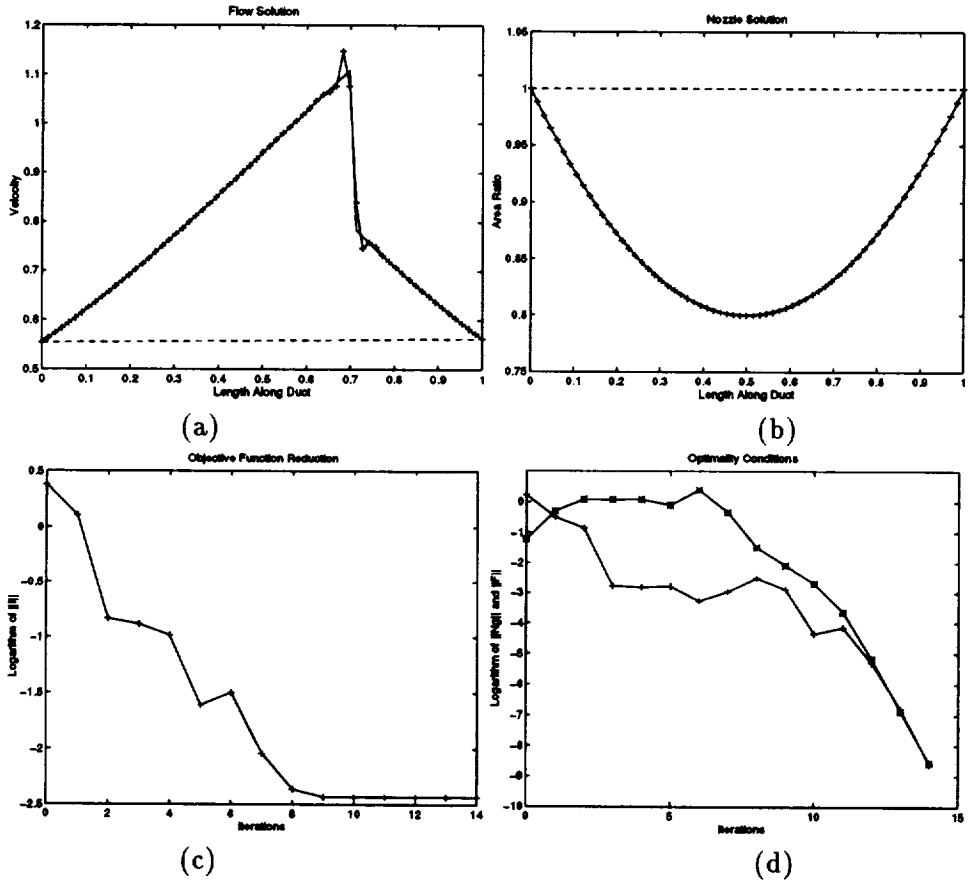


Figure 3: Computational results with 2 design variables. (a) Target (solid line), initial (dashed line) and final (checked solid) flow solutions. (b) Target (solid line), initial (dashed line) and final (checked solid line) nozzle profiles. (c) Reductions in objective function. (d) Reduction in projected gradient (+) and flow residual (*).

We suspect that this is partially due to the fact that an intermediate partial nozzle solution actually helps the flow solver to locate the shock and to estimate its magnitude.

Cost comparisons. Finally, we discuss the performance of the reduced Hessian SQP scheme with respect to the pure analysis (solution of nonlinear flow equations). Table 1 indicates that optimization with two design variables is only slightly more expensive than the analysis (costing 1.37 nonlinear flow solves). For the same problem, a brute force finite difference approach would require 3 nonlinear flow solves for a single optimization iteration. To solve the design problem, the number of nonlinear flow solves required by either an implicit gradient method or an adjoint equations approach would be at least twice as much as the number of optimization iterations which may easily exceed 10. For the situation with ten design variables, a fairly tough problem due to over-parameterization, the performance

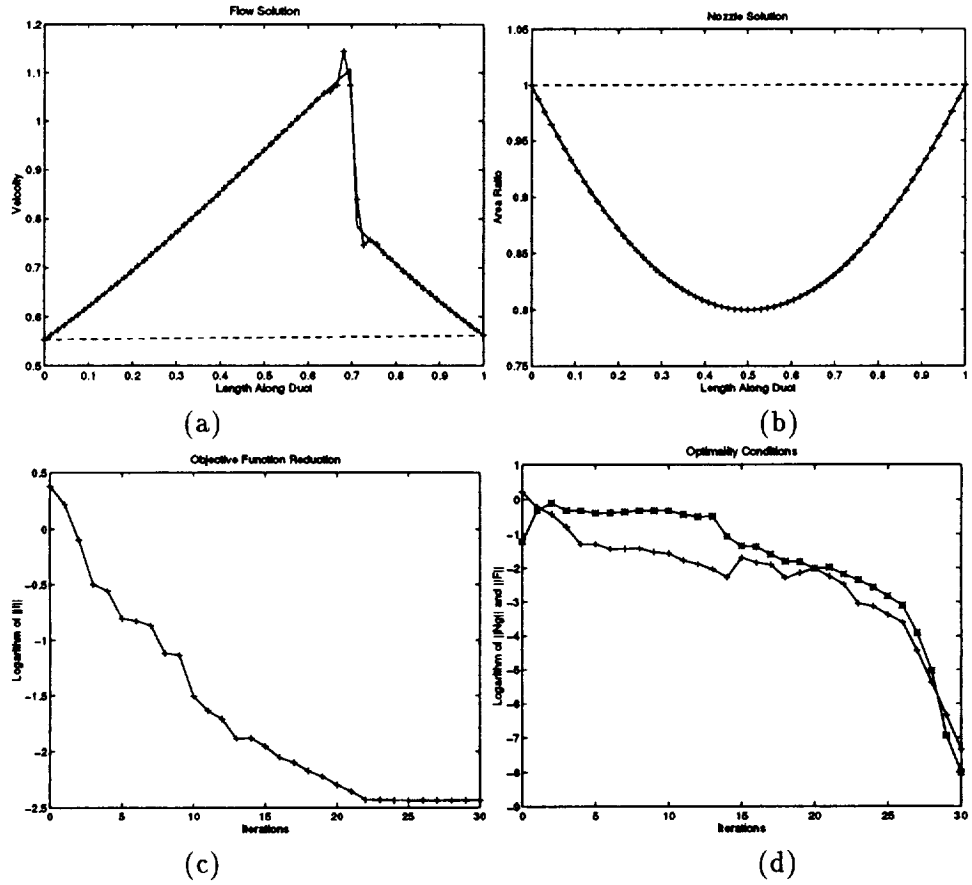


Figure 4: Computational results with 10 design variables. (a) Target (solid), initial (dashed) and final (checked solid) flow solutions. (b) Target (solid), initial (dashed) and final (checked solid) nozzle profiles. (c) Reductions in objective function. (d) Reduction in projected gradient (+) and flow residual (*).

of the reduced Hessian SQP scheme is still reasonably good. It costs an equivalent of 8.217 nonlinear flow solves, which would compare favorably over the other three methods mentioned above for similar reasons.

5 Concluding remarks

An efficient and robust reduced Hessian SQP scheme for aerodynamic design optimization has been introduced in this paper. This scheme requires neither the solution of an adjoint equation nor any exact second order information. The optimization was able to converge from tough starting points. Computational results show that this scheme has a potential to achieve the same order of cost as one nonlinear flow solve. However, much more work

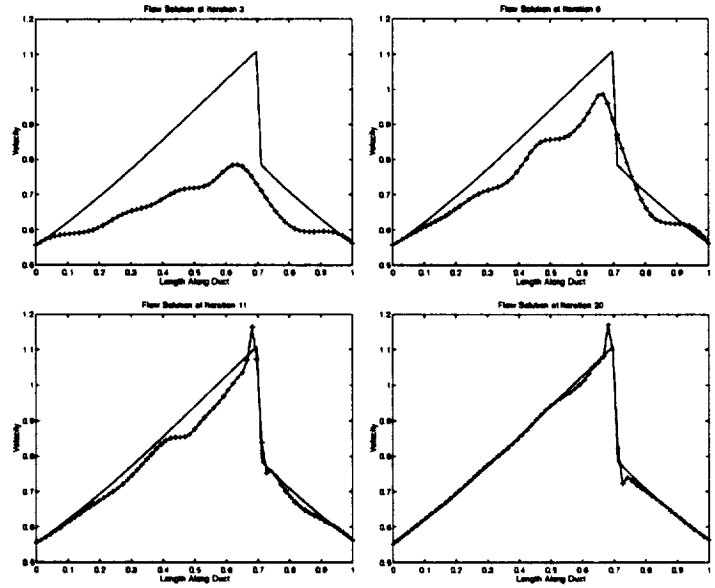


Figure 5: Snapshots of flow solutions at various iterations. Solid line – target solution; Checked solid line – current solution.

Number of Design Variables	Total Flops	Number of Iterations	Cost Ratio
0	2142176	20	1.000
2	2935839	14	1.370
10	17602144	30	8.217

Table 1: Efficiency of the all-at-once scheme

is still needed before we have a practical scheme. We believe the efficiency of this scheme can be improved if combined with grid sequencing [20] or multigrid techniques [15]. These techniques give efficient ways of providing a good starting point for the design on the final mesh. Issues such as dealing with inequality constraints and applications to 2D and 3D problems are currently under investigation.

References

[1] T. BARTH, *An unstructured mesh Newton solver for compressible fliud flow and its parallel implementation*, AIAA Paper 95-0221, (1995).

[2] L. T. BIEGLER, J. NOCEDAL, AND C. SCHMID, *A reduced Hessian method for large-scale constrained optimization*, Tech. Rep. CRPC-TR93432, Center for Research on Parallel Computation, Rice University, Houston, Texas, 1993.

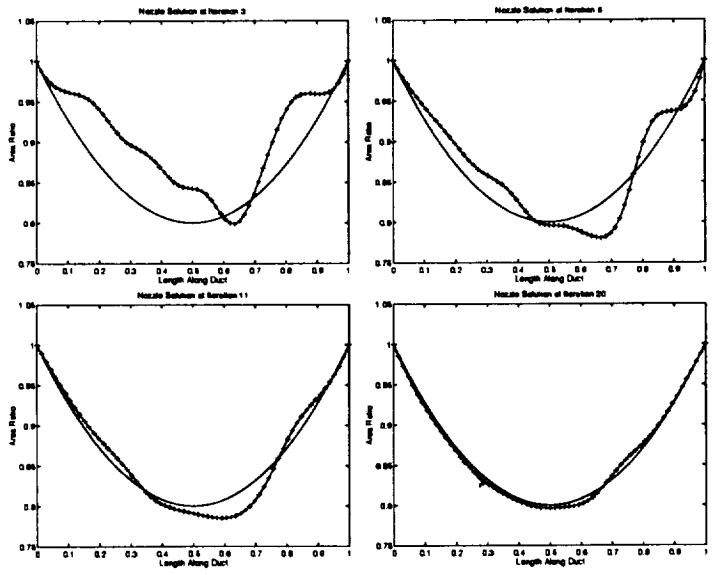


Figure 6: Snapshots of nozzle solutions at various iterations. Solid line – target solution; Checked solid line – current solution.

[3] R. H. BYRD AND J. NOCEDAL, *An analysis of reduced Hessian methods for constrained optimization*, Math. Programming, 49 (1991), pp. 285–323.

[4] T. F. COLEMAN AND A. CONN, *On the convergence of a quasi-Newton method for the nonlinear programming problem*, SIAM J. Numer. Anal., 21 (1984), pp. 755–769.

[5] J. E. DENNIS, M. HEINKENSCHLOSS, AND L. N. VICENTE, *Trust region interior-point algorithms for a class of nonlinear programming problems*, Tech. Rep. CRPC-TR95512, Center for Research on Parallel Computation, Rice University, Houston, Texas, 1995.

[6] R. FLETCHER, *Practical Methods of Optimization, Second Edition*, John Wiley & Sons, second ed., 1989.

[7] P. D. FRANK AND G. R. SHUBIN, *A comparison of optimization-based approaches for a model computational aerodynamics design problem*, Journal of Computational Physics, 98 (1992), pp. 74–89.

[8] D. GABAY, *Reduced quasi-Newton methods with feasibility improvement for nonlinearly constrained optimization*, Mathematical Programming Studies, 16 (1982), pp. 18–44.

[9] J. C. GILBERT, *Maintaining the positive definiteness of the matrices in reduced Hessian methods for equality constrained optimization*, Math. Programming, 50 (1991), pp. 1–28.

[10] G. H. GOLUB AND C. F. V. LOAN, *Matrix Computations*, The John Hopkins University Press, second ed., 1989.

[11] C. B. GURWITZ AND M. L. OVERTON, *Sequential quadratic programming methods based on approximating a projected Hessian matrix*, SIAM J. Sci. Stat. Comput., 10 (1989), pp. 631–653.

[12] G. W. HOU, A. C. TAYLOR, III, S. V. MANI, AND P. A. NEWMAN, *Formulation for simultaneous aerodynamic analysis and design optimization.* Manuscript, 1993.

[13] W. P. HUFFMAN, R. G. MELVIN, D. P. YOUNG, F. T. JOHNSON, J. E. BUSSOLETTI, M. B. BIETERMAN, AND C. L. HILMES, *Practical design and optimization in computational fluid dynamics*, AIAA Paper 93-3111, (1993).

[14] A. JAMESON, *Aerodynamic design via control theory*, Journal of Scientific Computing, 3 (1988), pp. 233–260.

[15] G. KURUVILA, S. TA'ASAN, AND M. D. SALAS, *Airfoil design and optimization by the one-shot method*, AIAA Paper 95-0478, (1995).

[16] J. NOCEDAL AND M. L. OVERTON, *Projected Hessian update algorithms for nonlinearly constrained optimization*, SIAM J. Numer. Anal., 22 (1985), pp. 821–850.

[17] T. PULLIAM, *Artificial dissipation models for the Euler equations*, AIAA Paper 85-0438, (1985).

[18] G. R. SHUBIN, *Obtain 'cheap' optimization gradients from computational aerodynamics codes*, Tech. Rep. AMS-TR-164, Boeing Computer Services, Seattle, Washington, 1991.

[19] Y. XIE AND R. H. BYRD, *Practical update criteria for reduced Hessian SQP, Part I: Global analysis*, Tech. Rep. CU-CS-753-94, Computer Science Department, University of Colorado, Boulder, 1994.

[20] D. P. YOUNG, W. P. HUFFMAN, M. B. BIETERMAN, R. G. MELVIN, F. T. JOHNSON, C. L. HILMES, AND A. R. DUSTO, *Issues in design optimization methodology*, Tech. Rep. BCSTECH-94-007 REV. 1, Boeing Computer Services, Seattle, Washington, 1994.

RIACS
Mail Stop T041-5
NASA Ames Research Center
Moffett Field, CA 94035